Git Basics with Android Studio

CS4084 - Mobile Application Development

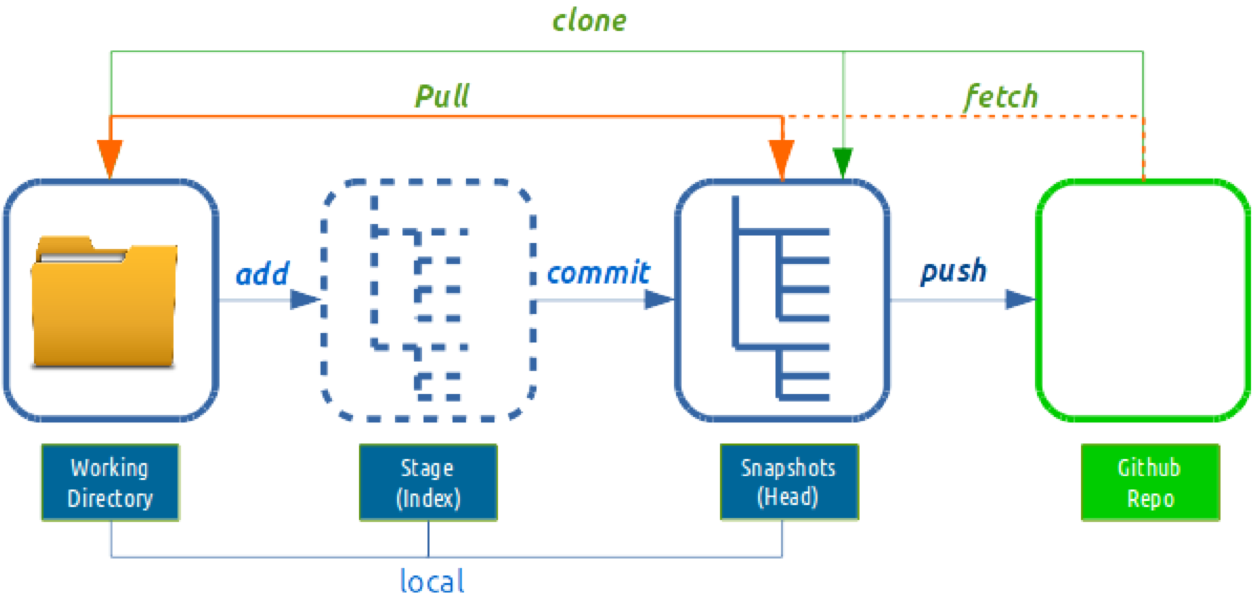
Habtom Gidey
University of Limerick

Git is a version control system (VCS). A VCS is a software source code change or revision management system that enables track of code commits by timestamp associated with the person who made the changes. Revision control is important element to establish effective configuration management system in software development with a team of contributors. Today, there are many version control systems, such as Mercurial, SVN (Subversion) and Bazaar. However, Git is popular for its strong design on distributed version control and efficient collaboration features.

Git is mainly used as a command line mode, here in this tutorial, we introduce the use of Android Studio as a git GUI client.

Fundamentals

The diagram below illustrates the basic concepts how a git versioning system works assuming GitHub as remote repository host:



Terminology:

- **Working Directory:**
    - *This contains the actual project files, this includes any not yet committed changes or edits made in your project files.*
- **GitHub Repo:**
    - *This is the remote repository in GitHub. GitHub is a source code hosting service mainly for git repositories. The web platform also provides additional services such as bug tracking, issue management, wikis and others.*
- **Commits:**
    - *Commits are checkpoints in git, checkpoints can be compared and restored. Each commit creates a snapshot of all the files added/staged to the git index, the changes on the edited files and the other files.*

- **Index (Stage):**
    - *The index contains staged/added files from the working directory to be tracked by the git version controller. Git only tracks changes with staged files on the git index. Files, such as build configuration files can be ignored from the versioning system. Put the files to be ignored in the .gitignore file which you can also locate inside your project directory.*
- **Head (snapshots)**:
    - *The head is the local repository that points to the committed snapshots.*
- **Pull and Push:**
    - *To pull is to request or fetch updates from and integrate with a repository. Push is a related command that enables to upload snapshots to a repository. It is always better to do a pull request before pushing your commits.*

Installing Git:

- https://git-scm.com/downloads (guide for Linux | Windows | Mac)

Git Basic Commands

If you are using the command line interface, here are the basic git commands to get you started:

1. *Setting up Git*

After installing git you can check the version of git with the following command and if Git is running you can set your identity and get started:

```
git --version
```

Configuration:

```
git config --global user.name "Carl Sagan"
git config --global user.email "carl.sagan@seti.org"
```

2. *Git Repository*

A repository can be created local or on a git repository server such as GitHub.

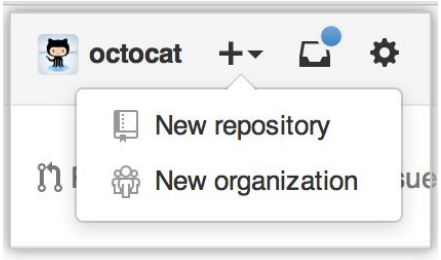
Local repository:

Change your directory to your working or project directory and initialize it as Git repository with init command:

```
cd ~/<projectDirectory>
git init
```

GitHub Repository:

Create an account in GitHub, once logged in click the plus menu at the top right of the page and select ‘***New repository***’



Cloning:
After creating a repository in GitHub you can clone or download a project and its entire version history to your workstation with the ***git clone*** command:

```
git clone <repo url>
```

```
git clone https://github.com/<username>/<repository>.git
```

3. *Staging:*

To add files from your *working directory* to git index, use the command **git add**:
(note the period, which indicates the *current directory*)

```
git add .
```

The above command adds all the files in your current directory to your git stage/index.

To add individual files separately use file names instead of the ‘dot’.
Use the asterisk wildcard to add files with specific extensions.

```
git add *.java
```

Related commands:
`git add -p, git add -i, git rm <filename>, …`

4. *Commit Changes*

To commit changes and create snapshots, use the command **git commit**. In git version control systems a commit creates snapshots of your working directory all the staged files including the files that are not changed.

```
git commit --m "project initial commit"
```

```
Related commands: git status, git diff, git log, …
```

5. *Push*

To update changes into your GitHub repository use:

```
git push
```

```
Related commands: git status, git diff, ...
```

6. *Pull*

To update changes from your GitHub repository use:

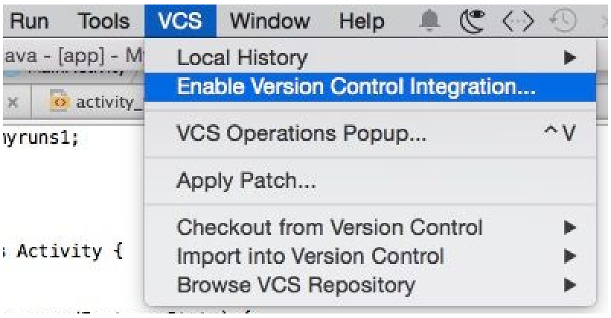
```
git pull

Related commands: git checkout, git fetch, git status, git log, ...
```

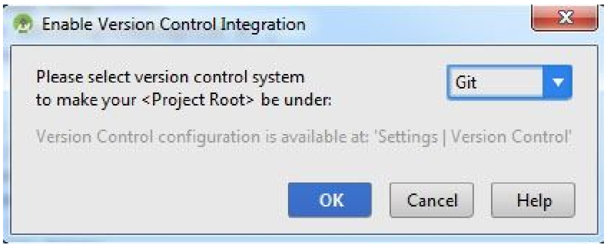
Git on Android Studio

Most of the git commands can also be done with Git GUI clients, Android Studio provides this features. Here are some of the git basics you can work with android studio:

1. Enable Git Integration

On the VCS menu of your Android Studio, you will find most of the commands to work with git. Go to ***VCS*** > ***Enable Version Control Integration*** and choose "Git" from the registered Version Control Systems in the Version Control Integration dialog box. This menu item and the dialog box is available only if the project does not already use any Version Control System.





Please select version control system to make your <Project Root> be under: Git

Version Control configuration is available at: 'Settings | Version Control'

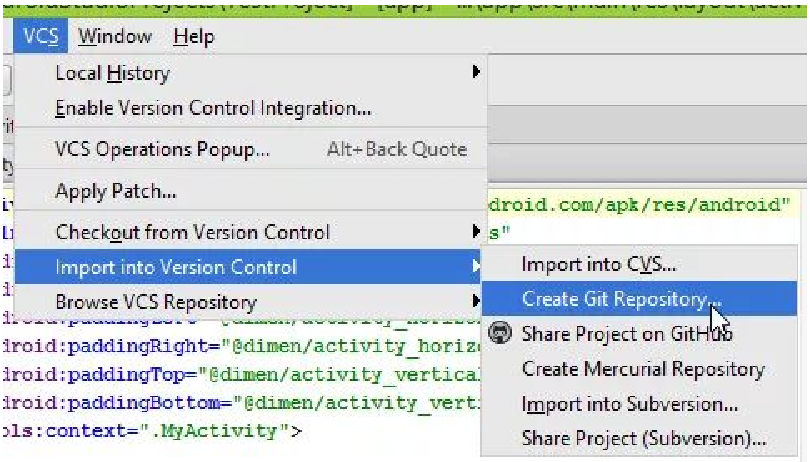
Configuration Note:

*Inside your project directory, usually located at **AndroidStudioProjects** folder, you will find the **.git** directory where the git **config** file is located. For a basic project level configuration **add your GitHub user account** as the following example, opening the file with a text editor:*

```
[user]
       name = Carl Sagan
       email = carl.sagan@seti.org
```

2. Sharing or cloning Repository on GitHub

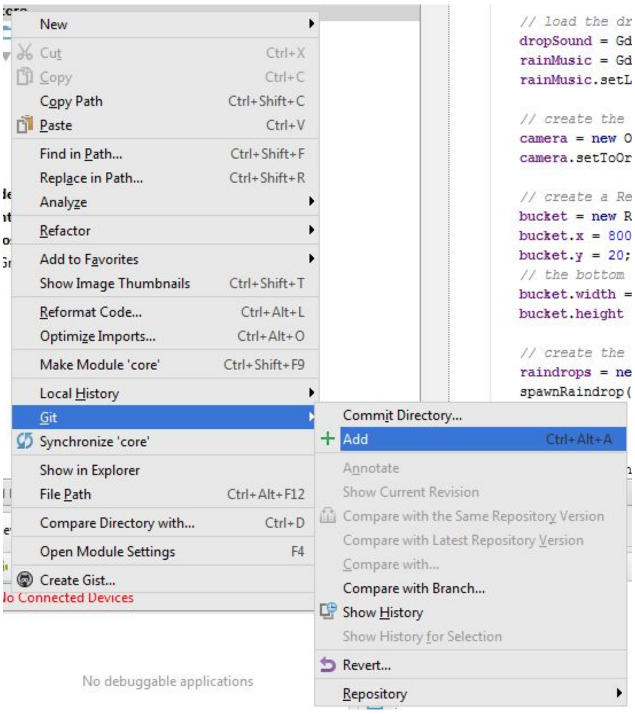
To clone from GitHub, go to ***CVS*** and select ***Checkout from Version Control*** and then provide your identity details and the repository you want to clone.


VCS
Window
Help
Local History
Enable Version Control Integration...
VCS Operations Popup... Alt+Back Quote
Apply Patch...
Checkout from Version Control
Import into Version Control
Browse VCS Repository
Import into CVS...
Create Git Repository...
Share Project on GitHub
Create Mercurial Repository
Import into Subversion...
Share Project (Subversion)...

To share on GitHub, similar way, go to ***VCS*** menu and select ***Import into Version Control*** and then ***Share Project on GitHub***. Provide your identity details and the repository you want to share. (VCS > Import into Version Control > Share Project on GitHub)

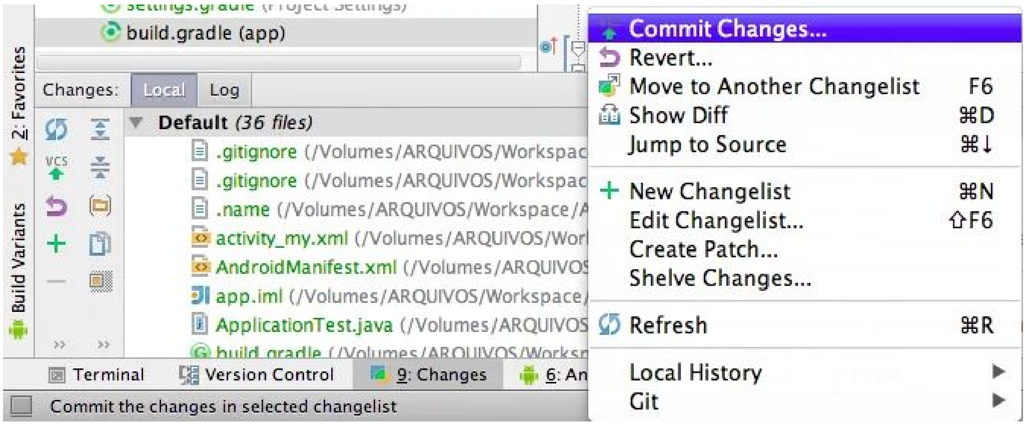
3. Stage - Add Files to git Index

To add or stage files, select the project or files you want to add to git index and go to ***VCS*** -> ***Git*** and select ***Add*** command from the menu item. You can similarly do this by right clicking on the project directory.



4. Commit Changes

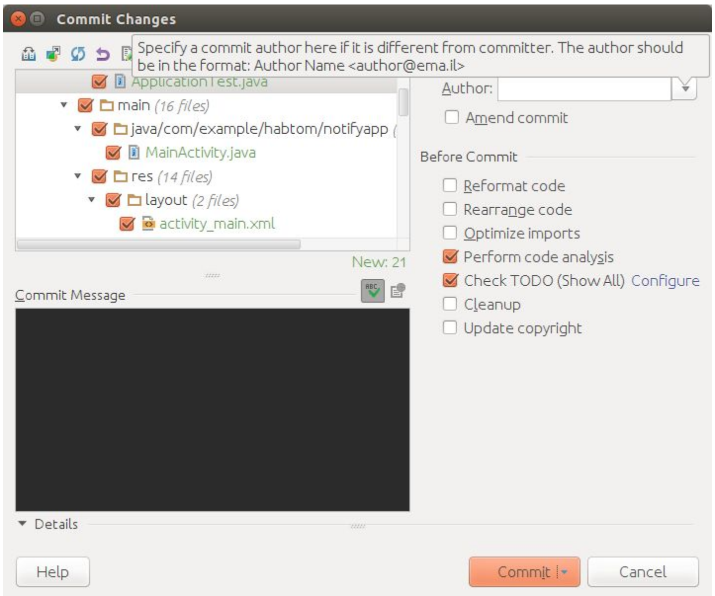
To commit changes, go to ***VCS*** and select ***Commit Changes*** from the menu item.



build.gradle (app)
Changes: Local Log
Default (36 files)
.gitignore (/Volumes/ARQUIVOS/Workspac
.gitignore (/Volumes/ARQUIVOS/Workspac
.name (/Volumes/ARQUIVOS/Workspace/A
activity_my.xml (/Volumes/ARQUIVOS/Wor
AndroidManifest.xml (/Volumes/ARQUIVOS
app.iml (/Volumes/ARQUIVOS/Workspace/
ApplicationTest.java (/Volumes/ARQUIVOS
2: Favorites
Build Variants
Terminal
Version Control
9: Changes
6: An
Commit Changes...
Revert...
Move to Another Changelist F6
Show Diff ⌘D
Jump to Source ⌘↓
New Changelist ⌘N
Edit Changelist... ⇧F6
Create Patch...
Shelve Changes...
Refresh ⌘R
Local History
Git

Commit the changes in selected changelist

In the ***Commit Changes*** dialog box, write down ***Commit Message***, then click ***Commit***. You can also select ***Commit and Push*** to immediately push your commit snapshots.



Specify a commit author here if it is different from committer. The author should be in the format: Author Name <author@ema.il>

Author:

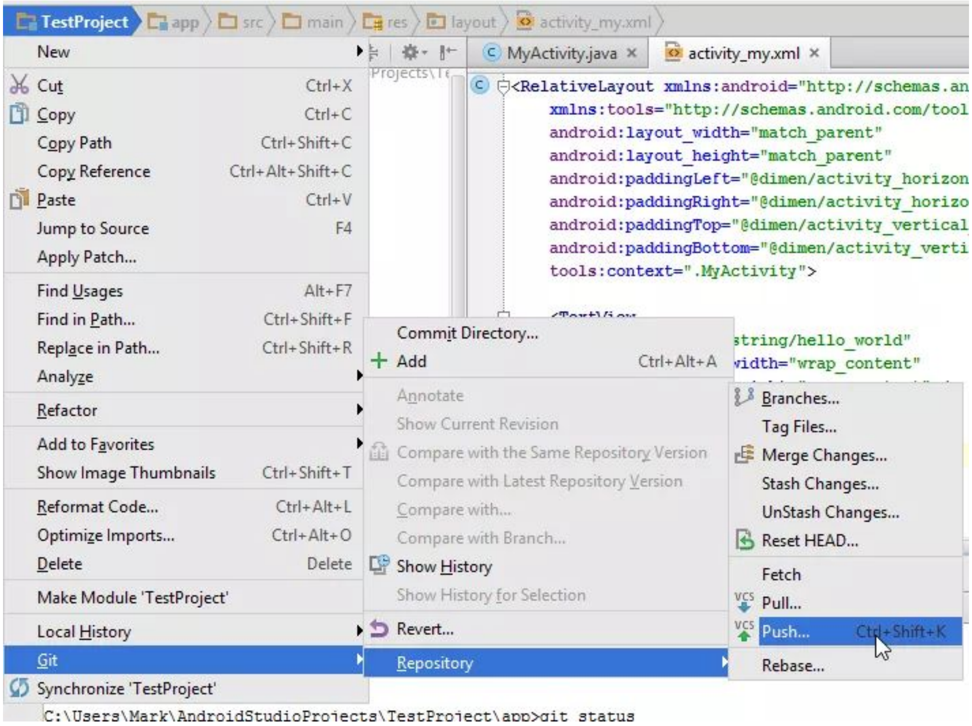
☐ Amend commit

Before Commit

☐ Reformat code
☐ Rearrange code
☐ Optimize imports
☑ Perform code analysis
☑ Check TODO (Show All) Configure
☐ Cleanup
☐ Update copyright

5. Push Committed Snapshots

To push commits, go to ***VCS*** -> ***Git*** and select ***Push*** from the menu item. You can similarly do this by right clicking on the project directory.



```
<RelativeLayout xmlns:android="http://schemas.an
    xmlns:tools="http://schemas.android.com/tool
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizon
    android:paddingRight="@dimen/activity_horizo
    android:paddingTop="@dimen/activity_vertical
    android:paddingBottom="@dimen/activity_verti
    tools:context=".MyActivity">
```

```
C:\Users\Mark\AndroidStudioProjects\TestProject\app>git status
```

6. Pull for updates

To pull changes into your local directory, go to ***VCS*** -> ***Git*** and select ***Pull*** from the menu item. You can similarly do this by right clicking on the project directory.

7. Adding contributors to a GitHub repository

To collaborate with your friends, you can begin with simple collaboration possibilities by allowing your teammates to contribute to a GitHub repository. On the GitHub website ***browse to your repository page*** and click on **Settings**. On the **Collaborators** page, you can type the collaborator's GitHub username which lets you select your teammates name to add as collaborators to your project and click ***Add collaborator***.